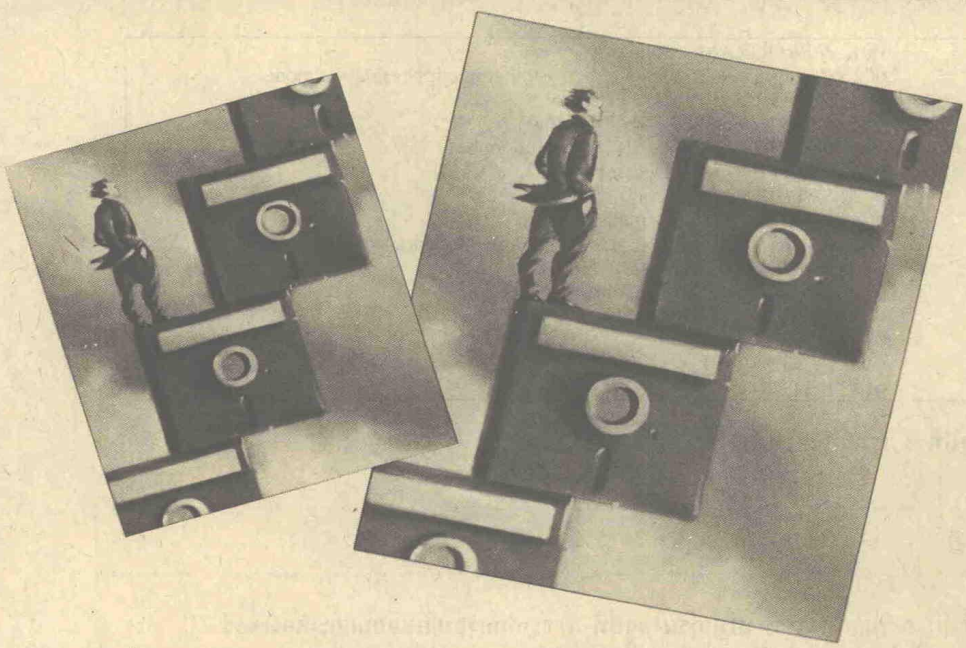
SPREAD SHEET

พงษ์ศักดิ์ หิรัณยรัชต์

ก้าวแรกในการสร้าง มาโครเอ็กเซล



อธิบายรายละเอียดต่าง ๆ ของการสร้างมาโคร ตั้งแต่ระดับขั้นพื้นฐาน จนถึงขั้นใช้สำหรับงานเฉพาะอย่าง

เป็นที่ทราบกันว่าซอฟต์แวร์ทางด้านสเปรดชีตทั้งรันบนดอสหรือบนวินโดวส์นั้นได้รับความนิยมในการใช้เป็นอันมาก เพราะว่าใช้งานง่าย เรียนรู้ง่าย ประยุกต์เข้ากับงานที่ทำอยู่ได้ง่าย และสุดท้ายคือ สามารถทำงานในโหมดอัตโนมัติได้ดีพอใช้ ในบ้านเราโลตัส 1-2-3 คือสเปรดชีตยอดนิยม ผู้เขียนมีโอกาสใช้ครั้งแรกเมื่อ 7 ปีที่แล้ว แต่ละฟังก์ชันของโลตัสทำให้ผู้เขียนร้องกรี๊ดด้วยความดีใจ ไม่เพียงแต่คำนวณเก่งเท่านั้น โลตัสยังสามารถสร้างกราฟประกอบการคำนวณได้ดีอีกด้วย โลตัสผงาดอยู่ในวงการสเปรดชีตอยู่นาน จนกระทั่งถึงยุคของวินโดวส์ โลตัสฟอร์วินโดวส์ก็มีออกมาให้พวกชอบของใหม่ได้ลองใช้กัน แต่การใช้งานยังถือว่าอยู่ในระดับไม่คล่องตัว

เมื่อกล่าวถึงไมโครซอฟต์เอ็กเซลเวอร์ชัน 3.0 แล้วผู้ใช้ระดับมืออาชีพต่างสรรเสริญแซ่ซ้องกันว่าเป็นซอฟต์แวร์สเปรดชีตระดับไร้เทียมทาน และในเวอร์ชัน 4.0 เอ็กเซลได้ยกระดับความเป็นมาตรฐานให้สูงขึ้นไป อีกทั้งขยายความสามารถในการใช้งาน และเพิ่มความคล่องตัวในการใช้ให้สูงขึ้นไปอีก อาจกล่าวได้ว่าไมโครซอฟต์เอ็กเซลเวอร์ชัน 4.0 เป็นซอฟต์แวร์เอนกประสงค์ไม่ว่าจะทำงานด้านการคำนวณ ฐานข้อมูล กราฟิก การออกรีพอร์ตงานด้านนำเสนอ (Presentations) และสุดท้ายคือมาโครที่มีความสามารถในการทำงานสูง

แนวทางของบทความ

ในบทความจะใช้ไมโครซอฟต์เอ็กเซล 4.0 Thai Edition เป็นต้นแบบในการอธิบายรายละเอียดต่าง ๆ โดยเริ่มตั้งแต่พื้นฐานสำหรับผู้ที่ยังไม่เคยสัมผัส

มาโครจนกระทั่งถึงการสร้างมาโครสำหรับงานเฉพาะอย่าง (Application) การสร้างมาโครสำหรับงานเฉพาะอย่างประกอบไปด้วยการสร้างเมนู การสร้างกรอบโต้ตอบ และการรวบรวมเข้าเป็นหน่วยเดียวกัน คำสั่งและฟังก์ชันต่าง ๆ อ้างอิงจากหนังสือ Microsoft Excel Function Reference

มาโครคืออะไร

มาโครคือชุดของคำสั่งที่นำคำสั่งต่าง ๆ ที่จะสั่งการไมโครซอฟต์เอ็กเซลมาเรียงกันเป็นคำสั่งใหม่ แล้วเรียกชื่อคำสั่งนั้นเสมือนเป็นเพียงคีย์เดียว ชุดคำสั่งดังกล่าวจะทำงานก็ต่อเมื่อมีการกดแป้นพิมพ์หรือเรียกชื่อคำสั่งนั้น ๆ มารันมาโครที่ง่าย ๆ อาจประกอบด้วยคำสั่งเพียงหนึ่งหรือสองคำสั่ง ในขณะที่มาโครที่ซับซ้อนอาจประกอบด้วยคำสั่งมากมาย อาจกล่าวโดยสรุปได้ว่า ในมาโครอาจประกอบด้วยคำสั่งเพียงหนึ่งหรือมากกว่าก็ได้

มาโครย่อมาจากคำว่า Macro-instruction เราใกล้ชิดกับมาโครมาเสียจนไม่รู้สึกตัว ไม่ว่าจะเป็นการเรียกใช้คำสั่งต่าง ๆ ผ่านพูลดาวน์เมนู หรือเรียกใช้ฟังก์ชันต่าง ๆ ภายใต้เมนู และฟังก์ชันก็คือมาโครนั่นเอง แต่เป็นมาโครที่เราไม่ได้เขียนขึ้นเอง (เราเรียกมาโครเหล่านี้ว่า "Internal Macros")

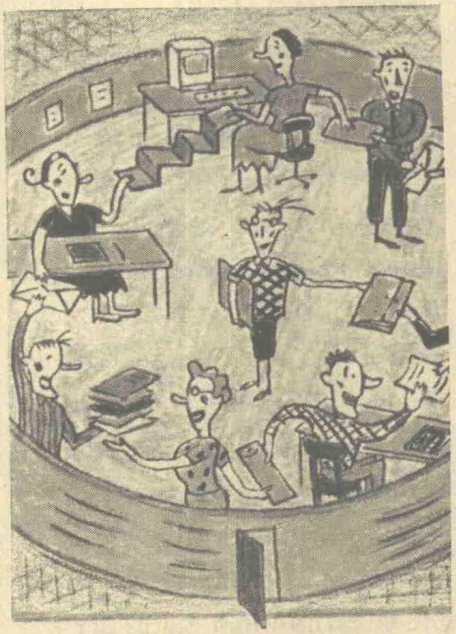
เราใช้มาโครทำอะไร

ก่อนเขียนมาโครเราควรไตร่ตรองสิ่งต่าง ๆ ให้รอบคอบเสียก่อน อย่างน้อยให้พิจารณาว่ามาโครที่กำลังจะเขียนขึ้นมาใช้งานสามารถทำงานที่ต้องการให้สำเร็จลุล่วงไปได้ด้วยดีได้หรือไม่ และต่อไปนี้คือตัวอย่างลักษณะงานที่มาโครสามารถทำงานได้ผลตามที่ต้องการ

● **สร้างฟังก์ชันที่ใช้บ่อย ๆ ขึ้นมาใช้เอง (Custom Functions)** คำสั่งและสูตรต่าง ๆ ที่อยู่บนมาโครชีตสามารถสร้างเป็นฟังก์ชันสำหรับงานเฉพาะอย่างตามที่เราต้องการได้ โดยมีประสิทธิภาพในการทำงานเหมือนกับฟังก์ชันที่ผู้พัฒนาโปรแกรมสร้างขึ้น (Build-in Function) ทุกประการ ทำให้เราไม่ต้องสร้างสูตรการคำนวณใหม่ทุกครั้ง

● **ทำงานที่มีขั้นตอนซ้ำซากหรือมีลักษณะเป็นงานที่ต้องทำเป็นประจำ** เช่น การออกรายงานประจำสัปดาห์ หรือรายงานประจำเดือน มาโครสามารถช่วยให้ท่านสร้างงานดังกล่าวได้อย่างรวดเร็วและถูกต้องแม่นยำขึ้นอีกด้วย

● **สร้างเมนูและกรอบโต้ตอบสำหรับงานเฉพาะด้าน (Applications)** ท่านสามารถสร้างมาโครสำหรับงานเฉพาะด้านได้เป็นอย่างดี งานเฉพาะด้านที่สร้างขึ้นอาจเป็นงานป้อนข้อมูลโดยที่ผู้ป้อนข้อมูลเป็นผู้ที่ไม่มีความรู้ด้านเวอร์กชีตแต่สามารถป้อนข้อมูลให้ท่านได้อย่างถูกต้องถ้ามีเมนูหรือกรอบโต้ตอบแนะนำ ทำให้ท่านประหยัดเวลาในการป้อนข้อมูลด้วยตนเองเป็นต้น



● **ทำให้สามารถรันแอพพลิเคชันอื่น ๆ ได้**

ชนิดของมาโครแบ่งตามลักษณะการใช้งาน

● **มาโครสำหรับงานยูทิลิตี้ (Utility Macros)** คือมาโครที่จัดว่ามีประโยชน์ต่องานที่ทำอยู่เป็นประจำทุกชนิด ไม่ว่าจะเป็นงานด้านเวอร์กชีตหรือแผนภูมิ

● **มาโครสำหรับงานเฉพาะอย่างย่อย ๆ (Task-Specific Macros)** คือมาโครสำหรับงานเฉพาะอย่างเล็ก ๆ เช่น มาโครสำหรับการหาผลรวมของยอดการผลิตประจำวัน เป็นต้น เรามักสร้างมาโครชนิดนี้เพื่อใช้งานเป็นส่วนใหญ่

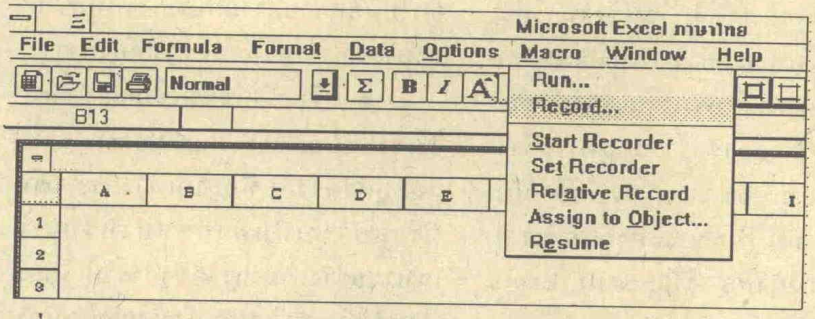
● **มาโครสำหรับงานเฉพาะอย่างเต็มรูปแบบ (Full Application Macros)** คือมาโครสำหรับงานเฉพาะอย่างที่สมบูรณ์แบบ มาโครชนิดนี้เป็นมาโครที่รวบรวมงานย่อย ๆ เข้าเป็นหน่วยเดียวกัน โดยใช้เมนู กรอบโต้ตอบ และทูลส์บาร์ เป็นเครื่องมือในการประสานงานย่อยเหล่านั้นให้ทำงานได้อย่างมีประสิทธิภาพ เช่น งานออกเลเบลลูกค้า งานออกใบสั่งซื้อ เป็นต้น

การสร้างมาโคร Global

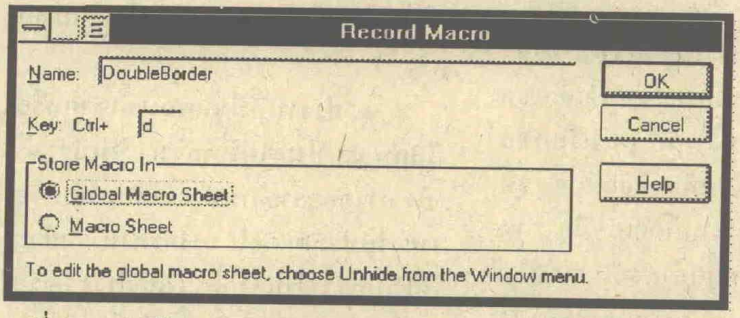
เพื่อให้เห็นภาพพจน์จะขอสร้างมาโครอย่างง่ายขึ้นมาก่อน หลักการสร้างและการรันโดยละเอียดจะกล่าวเป็นลำดับต่อไป การสร้างมาโครมีอยู่ด้วยกัน 2 วิธีคือ การบันทึก (Recor-

ding) ผ่านเมนู Macro ดังแสดงในรูปที่ 1 และการพิมพ์คำสั่ง (Editing) ผ่านแป้นพิมพ์ การบันทึกเป็นวิธีที่ง่ายที่สุดในการสร้างมาโคร ตัวบันทึกมาโครของไมโครซอฟต์เอ็กเซล (Microsoft Excel Macro Rocorder) จะทำการบันทึกการกดแป้นพิมพ์ (Keystokes) คำสั่งที่เรียกใช้สถานภาพของ dialog check box และอื่น ๆ เก็บเอาไว้ แต่จะไม่บันทึกการเคลื่อนที่ของเม้าส์ ไม่ต้องตกใจนะครับ แค่ไม่บันทึกการเคลื่อนที่ของเม้าส์เท่านั้นเอง แต่จะบันทึกผลลัพธ์ที่เกิดจากการคลิ๊กเม้าส์ เช่น การเลือกปุ่ม Option หรือการกำหนดขอบเขตหรือย่าน (Range) ให้กับเซล

ส่วนการสร้างมาโครโดยการพิมพ์คำสั่งลงไปในมาโครชีตนั้นยากกว่าวิธีแรก แต่ให้ความคล่องตัวมากกว่า โดยปกติเรามักใช้การสร้างมาโครด้วยการบันทึกเสียก่อนแล้วมาปรับแต่งรายละเอียดบางอย่างที่ต้องการภายหลัง



รูปที่ 1 การสร้างมาโครโดยการบันทึก (Recording) ผ่านเมนู Macro



รูปที่ 2 กรอบโต้ตอบ Record Macro

มาโครแรกของฉัน

ในหนังสือภาษาอังกฤษ กิจกรรมใดก็ตามที่เริ่มทำครั้งแรกเขามักใช้คำว่า My First... นำหน้าเสมอ ๆ เช่น เมื่อถึงตอนแนะนำการสร้างมาโครแรกให้กับผู้อ่าน เขาอาจใช้คำว่า My First Macro เพื่อไม่ให้เกิดความน้อยหน้าผู้เขียนขอใช้คำนี้มาเป็นชื่อนำสำหรับย่อยหน้านี้เสียเลย "มาโครแรกของฉัน" ให้ความรู้สึกเหมือนกับว่ากำลังย้อนกลับไปวันเก่า ๆ ในวัยประถมต้น

เข้าเรื่องกันดีกว่า เรากำลังจะสร้างมาโครที่สามารถตีกรอบด้วยเส้นคู่ล้อมรอบพื้นที่ (Range) ที่กำหนดไว้ โดยที่พื้นที่ที่กำหนดจะมีขนาดเล็กหรือใหญ่ขนาดไหนก็ได้ไม่จำกัด ต่อไปนี้คือขั้นตอนในการสร้างมาโครโดยการบันทึก

1. ให้โหลดไมโครซอฟต์เอ็กเซลลงสู่หน่วยความจำเพื่อการใช้งาน ถ้าท่านโหลดเรียบร้อยแล้วหน้าจอจะปรากฏเวอร์กชีตว่าง ๆ อยู่

2. เลือกขอบเขต (Range) ที่ต้องการสร้างกรอบล้อมรอบโดยการคลิ๊กเม้าส์แล้วลากไปตามต้องการ

3. ไปที่เมนู Macro ดังแสดงในรูป 1 แล้วเลือก Record... กรอบโต้ตอบ Record Macro จะปรากฏให้เห็น ดังรูปที่ 2

4. ที่กรอบ Name ให้พิมพ์ DoubleBorder เพื่อกำหนดชื่อให้กับมาโคร จากนั้นที่กรอบ Key: Ctrl+ ให้พิมพ์ d ดังแสดงในรูปที่ 2

5. ให้คลิ๊กเม้าส์ที่หน้าข้อความ Global Macro Sheet แล้วคลิ๊กปุ่ม OK ขณะนี้ที่ตำแหน่ง Status Bar ตรงมุมล่างซ้ายของจอจะปรากฏข้อความ "Recording" ซึ่งหมายความว่าขณะนี้พร้อมที่จะบันทึกมาโครแล้ว

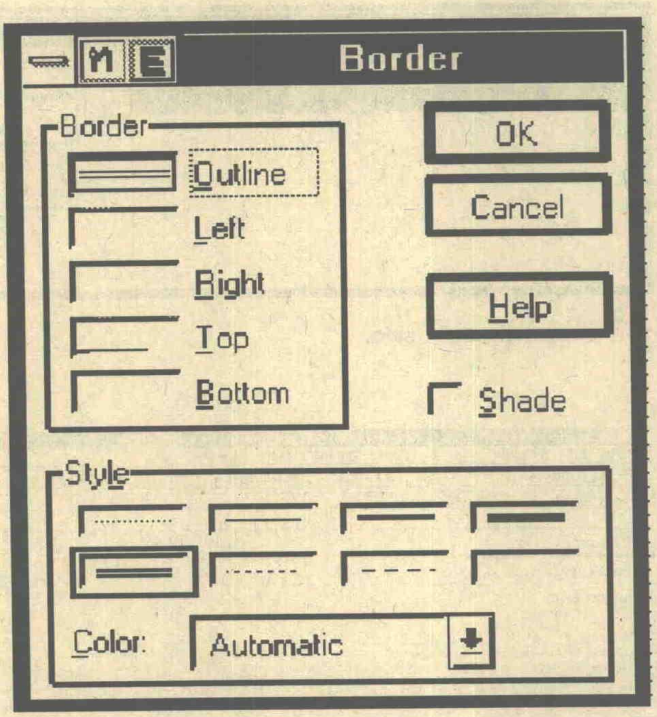
6. ไปที่เมนู Format เลือก Border กรอบโต้ตอบ Border จะปรากฏขึ้นมา

7. ที่สไตล์ให้เลือกแบบเส้นคู่ แล้วเลือก Outline ดังแสดงในรูปที่ 3

8. คลิ๊กปุ่ม OK ขณะนี้ขอบเขตที่ท่านเลือกไว้ตั้งแต่หัวข้อที่ 2 จะมีกรอบเส้นคู่ล้อมรอบ

9. ไปที่เมนู Macro เลือก Stop เพื่อยุติการบันทึกมาโคร

แล้วก็มาถึงเวลาชมผลงานที่ทำไว้ ให้คลิ๊กเม้าส์ไปที่นอกพื้นที่ที่กำหนดไว้ในหัวข้อที่ 2 จะเห็นว่ามีกรอบเส้นคู่ล้อมรอบพื้นที่ที่กำหนดไว้ มาโครใด ๆ ก็ตามที่สร้างบนมาโครชีตแบบ Global จะมีคุณสมบัติพิเศษคือ สามารถใช้งาน



รูปที่ 3
กรอบโต้ตอบ Border

หรือรันกับเวอร์กชีตใด ๆ ก็ได้ แตกต่างกับมาโครของโลตัส 1-2-3 ที่ใช้งานได้เฉพาะเวอร์กชีตที่เก็บมาโครไว้ ยกเว้นเสียแต่ว่าทำเป็นมาโครไลบรารี (Macro Library) จุดนี้เป็นข้อแตกต่างที่เห็นได้ชัดว่ามาโครของไมโครซอฟต์เอ็กเซลคล่องตัวกว่า

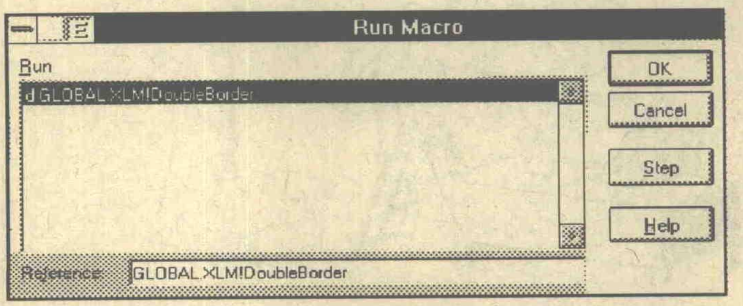
การรันมาโคร

เราสามารถรันคำสั่งมาโครได้หลายวิธีด้วยกัน

1. รันผ่านคำสั่งรัน (Run) ภายใต้เมนู Macro
2. รันผ่านแป้นพิมพ์ที่เป็น Shortcut Key
3. รันโดยการคลิ๊กเม้าส์ที่ Button (คือปุ่มพิเศษที่สร้างขึ้น ลักษณะคล้ายกับปุ่ม OK) หรือ Tool โดยที่ภายใต้บัททอนหรือทูลส์จะต้องมีคำสั่งมาโครฝังเอาไว้ จะกล่าวถึงการสร้างบัททอนและทูลส์โดยละเอียดในโอกาสต่อไป
4. รันโดยผ่านมาโครทูลส์บาร์ จะขออธิบายสองวิธีแรกเสียก่อน

การรันมาโครโดยใช้คำสั่ง Run... ภายใต้เมนูมาโคร

1. จากเมนู Macro เลือกคำสั่ง Run... กรอบโต้ตอบ Run Macro จะปรากฏดังแสดงในรูปที่ 4
2. ที่กรอบ Run ให้เลือกมาโครที่ต้องการรัน ในที่นี้คือ DoubleBorder หรืออาจใช้วิธีพิมพ์ชื่อมาโครที่ต้องการลงไปที่กรอบ Reference



รูปที่ 4 กรอบโต้ตอบ Run Macro

3. คลิ๊กปุ่ม OK

การรันมาโครโดยใช้ Shortcut Key

● ในกรณีนี้ให้กดแป้นเป็น Ctrl ค้างไว้ แล้วตามด้วยการกดแป้น d

ตามหามาโคร

ท่านที่เคยใช้โลตัสในการสร้างมาโครคงไม่ต้องตามหามาโคร เพราะมาโครจะอยู่ในเวอร์กชีตที่เปิดใช้งานอยู่ ยิ่งกรณีสร้างมาโครโลตัสด้วยวิธีบันทึกเราแทบไม่ต้องตามหาคำสั่งมาโครเลย เพราะก่อนบันทึกเราต้องกำหนดนิวาสถานให้มาโครโลตัสเสียก่อน ในขณะที่มาโครของเอ็กเซลจะถูกเก็บแยกไว้บนมาโครชีตต่างหาก ไม่ปะปนกับข้อมูลที่อยู่ในเวอร์กชีต ประโยชน์ในการเก็บคำสั่งมาโครไว้ต่างหากก็คือไมโครซอฟต์เอ็กเซลเปิดโอกาสให้เราโหลดมาโครชีตเพื่อใช้งานได้ครั้งละหลาย ๆ ไฟล์หรือครั้งละหลาย ๆ มาโครชีตนั่นเอง ประโยชน์ข้อที่สองก็คือเราสามารถแก้ไขมาโครชีตที่อยู่ในหน่วยความจำได้ครั้งละหลาย ๆ ไฟล์โดยไม่ต้องออกจากเอ็กเซล (Exit) เสียก่อน นี่คือความคล่องตัวของเอ็กเซลอีกประการหนึ่ง

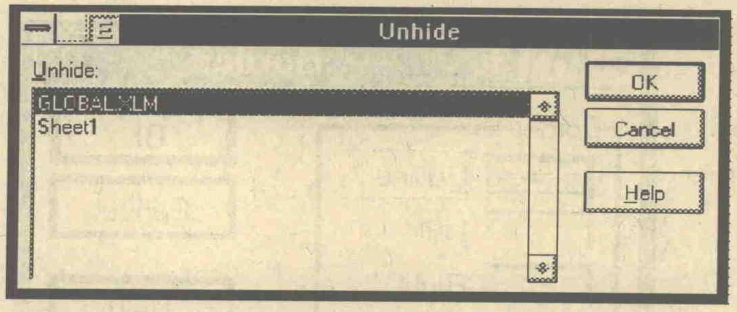
จะเห็นได้ว่าในเวอร์กชีตที่เปิดใช้งานอยู่ไม่มีคำสั่งมาโครแสดงอยู่เลย จึงต้องมีการตามหามาโคร

ตามไปดู

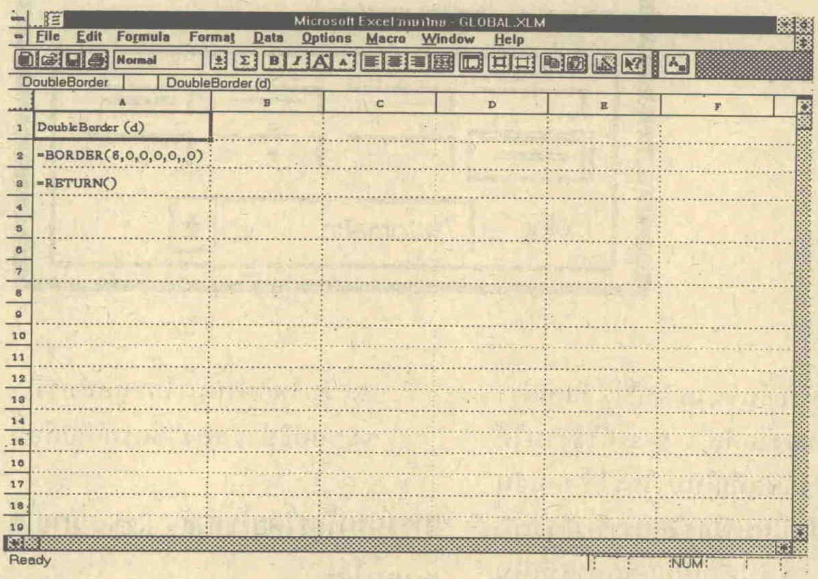
การสร้างคำสั่งมาโครบนมาโครชีตแบบ Global สิ่งที่เกิดขึ้นโดยอัตโนมัติอีกประการหนึ่งก็คือ เอ็กเซลจะทำการเก็บคำสั่งต่าง ๆ ไว้บนมาโครชีตชื่อ GLOBAL.XLM ซึ่งในสภาวะปกติจะมีสถานะภาพมองไม่เห็น (Hidden Macro Sheet) ต้องสั่งให้แสดงตัวจึงจะมองเห็น ขั้นตอนการทำให้แสดงตัวมีดังนี้

1. ไปที่เมนู Window เลือก Unhide

2. กรอบโต้ตอบ Unhide จะปรากฏ ให้เลือก GLOBAL.XLM แล้วคลิก OK (M ตัวสุดท้ายที่ปรากฏตรงส่วนนามสกุล ย่อมาจาก Macro) ดังรูปที่ 5 และ 6 ตามลำดับ



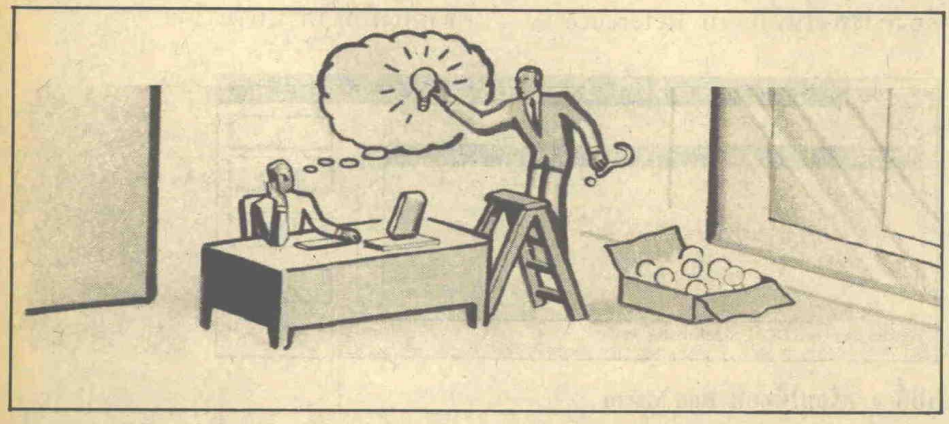
รูปที่ 5 กรอบโต้ตอบ Unhide



รูปที่ 6 มาโครชีตแบบ Global

เกร็ดเกี่ยวกับมาโครชีต Global

1. เมื่อสร้างมาโครถ้าเป็นไปได้ควรเก็บแบบ Global โดยเฉพาะอย่างยิ่งมาโครสำหรับงานยูทิลิตี้ที่ไม่ซับซ้อน เช่น มาโครสำหรับการเซฟไฟล์ หรือมาโครสำหรับพิมพ์รายงานประจำวัน เป็นต้น เราไม่จำเป็นต้องจำชื่อมาโครไฟล์ ไม่ต้องคอยมาเปิดไฟล์มาโครทุกครั้งที่ใช้งาน และประการสุดท้าย ไม่ต้องกังวลว่าได้เซฟไฟล์มาโครหลังจากการแก้ไขแล้วหรือยัง เพราะไมโครซอฟต์เอ็กเซลจะจัดการสิ่งเหล่านี้ให้เอง และเรายังสามารถรันมาโครที่เก็บแบบนี้ได้ทุกเวลาที่ต้องการอีกด้วย



2. การซ่อนมาโครชีตและการเซฟมาโคร เนื่องจากมาโคร Global มีค่าดีฟอลต์เป็นแบบซ่อน ดังนั้นเมื่อจะทำการแก้ไขหรือปรับปรุงจะต้องเปลี่ยนสถานะภาพให้สามารถมองเห็นได้ แต่หลังจากทำการแก้ไขเสร็จแล้ว อาจมีการลืมซ่อนอีกครั้งทำให้มาโครชีตมองเห็นตลอดเวลา อาจไม่ปลอดภัยจากมือใหม่ซึ่งอาจหวังดีช่วยแก้ไขให้เจริญลง ดังนั้นควรซ่อนมาโคร Global ทุกครั้งหลังจากแก้ไขหรือปรับปรุงเรียบร้อยแล้ว ขั้นตอนการซ่อนก็คือไปที่มาโคร-

ชีต Global แล้วเลือกคำสั่ง Hide จากเมนู Window การซ่อนมาโครเป็นอันเสร็จพิธี ส่วนการเซฟนั้นเมื่อเราออกจากไมโครซอฟต์เอ็กเซลด้วยคำสั่ง Exit จากเมนู File เอ็กเซลจะถามว่าต้องการเซฟมาโคร Global หรือไม่ ถ้าต้องการให้ตอบ Yes

3. แหล่งที่เก็บมาโคร Global ไมโครซอฟต์เอ็กเซลเก็บไฟล์ GLOBAL.XIM ไว้ที่ไดเรคทอรีชื่อ Startup

ความแตกต่างระหว่างเวอร์กชีตกับมาโครชีต

ข้อแตกต่างระหว่างเวอร์กชีตกับมาโครชีตที่เห็นได้ชัดก็คือความกว้างคอลัมน์ของมาโครชีตจะกว้างกว่าเวอร์กชีต ที่เป็นเช่นนี้เป็นเพราะว่าที่มาโครชีตสิ่งที่เก็บคือสูตรการคำนวณหรือสมการคณิตศาสตร์ ส่วนใหญ่มักมีความยาวจึงต้องใช้ความกว้างมากกว่าเวอร์กชีตซึ่งมักเก็บค่าที่เป็นตัวเลข (Value) เสียเป็นส่วนมาก ความกว้างของคอลัมน์ในมาโครชีตเป็นค่าดีฟอลต์

รู้นะคิดอะไรอยู่

การตรวจสอบค่าที่เก็บในมาโครทำให้เรารู้ว่ามาโครที่เราเขียนขึ้นเพื่อใช้งานมีขั้นตอนการทำงานถูกต้องตามที่ต้องการหรือไม่ วิธีการมีดังนี้

1. ไปที่มาโครชีต GLOBAL.XLM

2. ถ้าต้องการให้แสดงค่าให้กดแป้น Ctrl+` (แป้น ` อยู่ใต้แป้น Esc) จะได้ผลลัพธ์ดังแสดงในรูปที่ 7

3. ถ้าต้องการให้แสดงสูตรเหมือนก่อนหน้านี้กดแป้น Ctrl+` อีกครั้ง

การตรวจสอบชื่อมาโครที่อยู่บน มาโครชีต

คงจำกันได้ว่ามาโคร DoubleBorder ที่เราเพิ่งสร้างได้รับการกำหนดชื่อหรือตั้งชื่อไว้แล้ว จึงไม่จำเป็นต้องตั้งกันใหม่อีก แต่คราวนี้เราจะทำการตรวจสอบดูว่าชื่อที่ตั้งไว้มีอยู่ในมาโครชีตจริงหรือไม่ ขั้นตอนมีดังนี้

1. ไปที่มาโครชีต GLOBAL.XLM

2. ที่เมนู Formular เลือก Define Name จะปรากฏกรอบโต้ตอบ Define Name ขึ้นที่หน้าจอ

3. ที่กรอบแสดงรายการ (List box) Name in Sheet ให้คลิ๊กที่ DoubleBorder มาถึงขั้นนี้หน้าจอของท่านจะคล้ายกับรูปที่ 8

4. คลิกปุ่ม OK

ในกรอบโต้ตอบ Define Name ท่านจะพบว่า

1. ตำแหน่งของมาโครชื่อ DoubleBorder อยู่ที่ A1

2. ถูกจัดอยู่ในประเภท (Category) มาโครที่ผู้ใช้สร้างขึ้นเอง (User Defined)

3. มี Ctrl+d เป็น Shortcut Key

4. ที่กรอบแสดงรายการ (Name in Sheet) มีชื่อเรคอร์ดปรากฏอยู่ มาโครที่สร้างจากการบันทึก (Record) ผ่านเมนู Macro จะมีชื่อนี้ปรากฏอยู่เสมอ ทำหน้าที่กำหนดตำแหน่งที่อยู่ให้กับคำสั่งมาโครที่เกิดจากการบันทึก

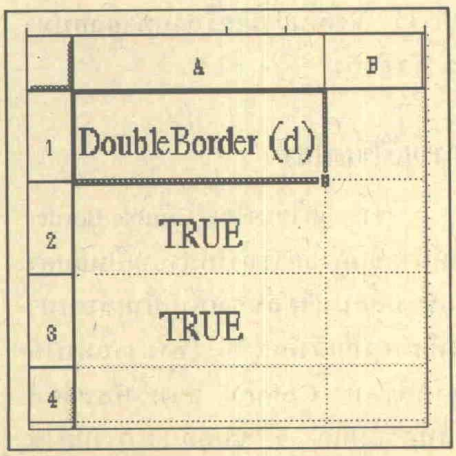
ถอดรหัสมาโคร DoubleBorder

จะขออธิบายความหมายของมาโครไปทีละบรรทัดเพื่อความเข้าใจร่วมกัน

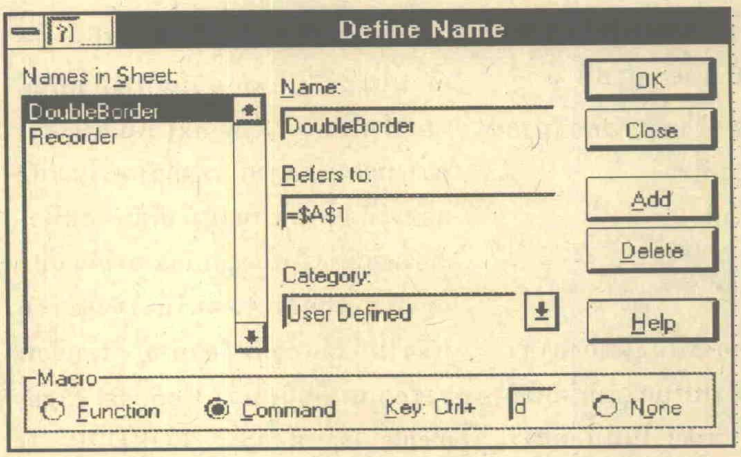
บรรทัดที่ 1

```
DoubleBorder(d)
```

บรรทัดนี้เป็นชื่อมาโคร ไมโคร-



รูปที่ 7 ค่าที่อยู่ในมาโครชีต



รูปที่ 8 กรอบโต้ตอบ Define Name

ซอฟต์เอ็กเซลจัดการสร้างให้โดยอัตโนมัติ เนื่องจากเอ็กเซลไม่ต้องการให้บรรทัดแรกของชุดคำสั่งมาโครมีเครื่องหมายเท่ากับ (=) นำหน้าจึงจัดการส่วนนี้ให้ แต่ก็สามารถลบหรือแก้ไขได้ ชื่อ DoubleBorder คือชื่อมาโคร ส่วน (d) เป็น Shortcut Key

บรรทัดที่ 2

```
=BORDER(6, 0, 0, 0, 0,, 0)
```

บรรทัดนี้เป็นการบอกให้เอ็กเซลทำการตีกรอบด้วยเส้นคู่ล้อมรอบพื้นที่ที่กำหนดไว้ รายละเอียดของคำสั่งมีดังนี้

```
BORDER (outline, left, right, top, bottom, shade, outline__color, left__color, right__color, top__color, bottom__color)
```

โดยที่ outline, left, right, top, bottom มีค่าเป็น 0 ถึง 7 ความหมายของรหัสแสดงอยู่ที่ตารางที่ 1 การใช้งานเพียงแต่ใส่รหัสที่ต้องการให้ตรงกับตำแหน่งของอาร์กิวเมนต์ เช่น ต้องการตีเส้นใต้แบบเส้นไข่ปลาที่ตำแหน่งอาร์กิวเมนต์ Bottom ให้ใส่รหัส 3 เป็นต้น

ส่วน outline__color, left__color, right__color, top__color, bottom__color มีค่าตั้งแต่ 0 ถึง 16 สอดคล้องกับสีที่แสดงอยู่ที่กรอบ Color ในกรอบโต้ตอบ Border ดูรูปที่ 3

บรรทัดที่ 2 จึงสามารถอธิบายรายละเอียดได้ดังรูปที่ 9

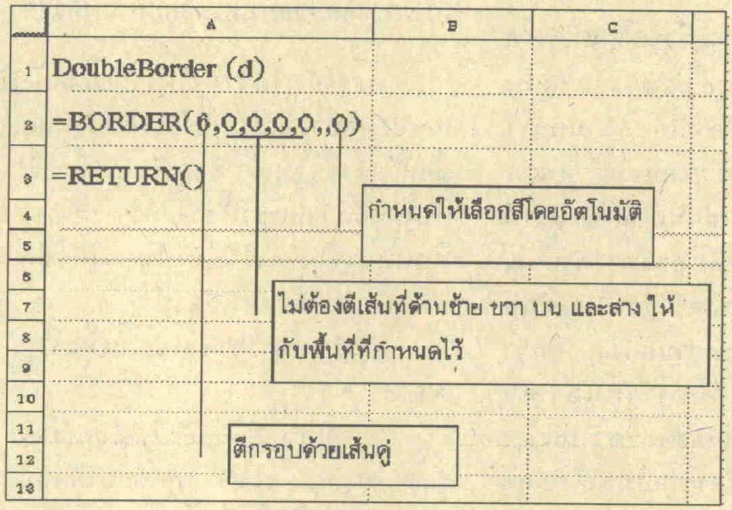
บรรทัดสุดท้าย

```
=RETURN( )
```

เป็นการบอกการสิ้นสุดของคำสั่งมาโครและกลับไปยังโปรแกรมที่เรียกใช้ (Caller) Double Border ในที่นี้คือการกลับไปสู่ไมโครซอฟต์เอ็กเซลนั่นเอง

รหัส	สไตล์ของเส้นกรอบ
0	ไม่ตีเส้น
1	เส้นบาง
2	เส้นปกติ
3	เส้นไข่ปลา
4	เส้นจุด
5	เส้นหนา
6	เส้นคู่
7	เส้นแฮร์ไลน์

ตารางที่ 1 รหัสและความหมายเส้นกรอบ (Border)



รูปที่ 9 ความหมายของมาโครบรรทัดที่ 2

อาร์กิวเมนต์ (Arguments)

คือตัวแปรที่อยู่ในฟังก์ชัน มักเขียนในวงเล็บ เช่น x เป็นอาร์กิวเมนต์ของฟังก์ชัน y = tan(x) เป็นต้น ตัวอย่างที่ยกมาแสดงเป็นฟังก์ชันตรีโกณมิติ แต่อาร์กิวเมนต์สำหรับฟังก์ชันคอมพิวเตอร์แตกต่างกันบ้างเล็กน้อย อาร์กิวเมนต์ของฟังก์ชันคอมพิวเตอร์บางตัวจะระบุหรือไม่ก็ได้แล้วแต่ต้องการ เราเรียกอาร์กิวเมนต์ชนิดนี้ว่า Optional Arguments จากฟังก์ชัน BORDER ณ ตำแหน่ง shade ซึ่งเป็น Optional Arguments ไม่ได้ระบุค่าอาร์กิวเมนต์เอาไว้ จึงเห็นว่ามีเครื่องหมายคอมมา 2 ตัวติดกัน

การแก้ไขมาโคร

เราจะแก้ไขมาโคร Double Border ให้น่าดูขึ้นด้วยการทำให้กรอบเป็นสีแดงและเปลี่ยนสไตล์ของเส้นเป็นเส้นหนา รหัสของสีแดงคือ 3 (สามารถนับได้จากกรอบ Color) ส่วนรหัสสไตล์เส้นกรอบคือ 5 ขั้นตอน การแก้ไขมาโครมีดังนี้

1. ไปที่มาโครชีต GLOBAL.XLM

2. คลิ๊กที่เซล A2 แล้วกดปุ่ม F2 เพื่อทำการแก้ไข

3. แก้ไขให้เป็นไปตามรูปที่ 10

ทดสอบผลลัพธ์ของมาโครที่เพิ่งแก้ไข ขั้นตอนมีดังนี้

1. กลับไปที่เวอร์กชีต โดยไปที่เมนู Window เลือกเวอร์กชีตที่ต้องการ

2. เลือกพื้นที่ที่ต้องการตีกรอบล้อมรอบ

3. กด Ctrl + d

4. คลิ๊กเม้าส์นอกบริเวณที่เลือกไว้ในข้อที่ 2

กรอบที่ได้มีสีสันขึ้นแล้ว

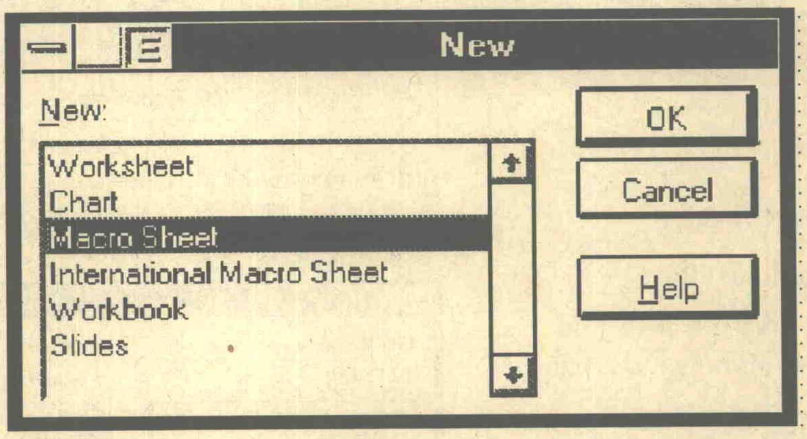
การสร้างมาโครบนมาโครชีต

ที่ผ่านมาเป็นการสร้างคำสั่งมาโครบนไฟล์ GLOBAL.XLM ต่อไปนี้จะเสนอการสร้างมาโครอีกประเภทหนึ่ง การสร้างคำสั่งมาโครมีหลักการเหมือนกับการสร้างเหมือนกับแบบ Global ต่างกันตรงที่จะต้องจัดเก็บไว้ในมาโครชีต นามสกุลของไฟล์ก็ยังคงเหมือนเดิมคือ .XLM

มาโครที่จะสร้างนี้เป็นมาโครสร้างเสียง Beep จึงตั้งชื่อว่า Beeper ขั้นตอนมีดังนี้

	A	B
1	DoubleBorder (d)	
2	=BORDER(5,0,0,0,0,,3)	
3	=RETURN()	

รูปที่ 10 มาโคร DoubleBorder ที่แก้ไขแล้ว



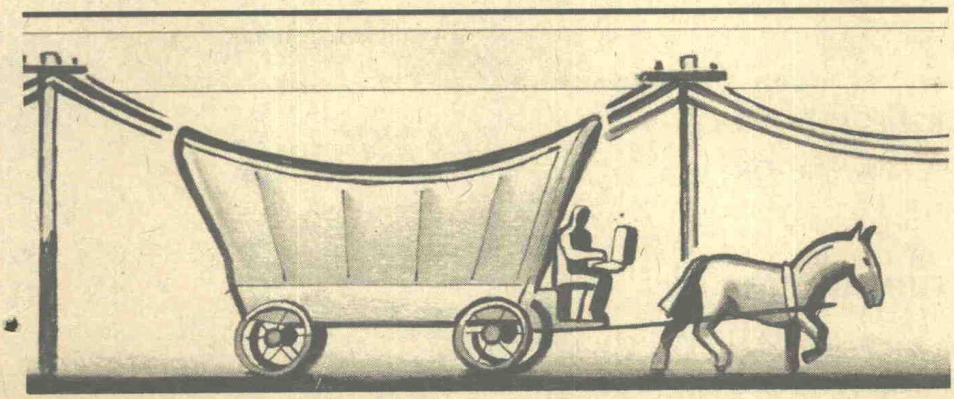
รูปที่ 11 กรอบโต้ตอบ New

1. ไปที่เมนู File เลือก New กรอบโต้ตอบ New จะปรากฏขึ้นมา ให้เลือก Macro Sheet แล้วคลิ๊ก OK ดังแสดงในรูปที่ 11

2. พิมพ์คำสั่งมาโครตามที่แสดงในรูป 12 ในการพิมพ์คำสั่งมาโครควรพิมพ์ด้วยตัวพิมพ์เล็ก ทันทีที่เคาะแป้น

	A
1	Beeper(b)
2	=BEEP()
3	=RETURN()
4	

รูปที่ 12 มาโคร Beeper



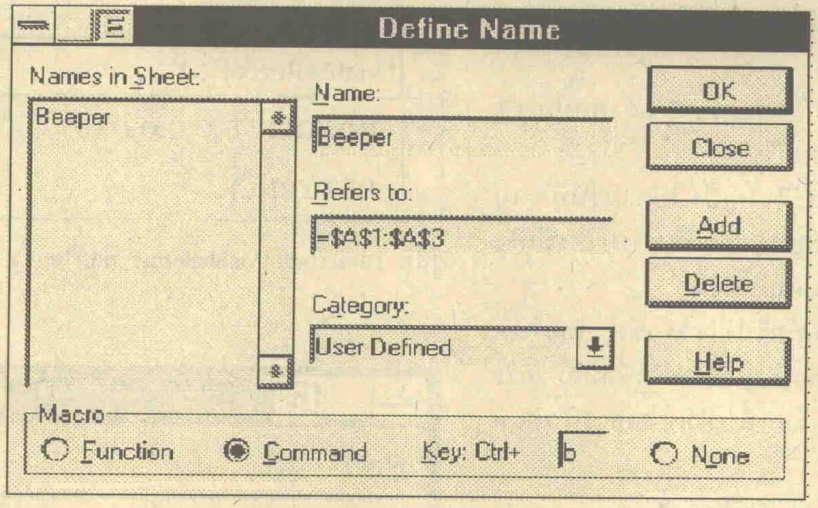
Enter (เมื่อสิ้นสุดคำสั่ง) เอ็กเซลจะทำการตรวจสอบความถูกต้องของคำสั่ง (Syntax) ถ้าคำสั่งถูกต้องเอ็กเซลจะเปลี่ยนคำสั่งเป็นตัวพิมพ์ใหญ่ให้โดยอัตโนมัติ ในทางกลับกันคำสั่งที่ไม่ถูก

เปลี่ยนเป็นตัวพิมพ์ใหญ่แสดงว่ามีบางอย่างผิดปกติ

3. คลิ๊กเม้าส์เพื่อเลือกช่วง (Range) A1:A3 จากนั้นไปที่เมนู Formula เลือก Define Name... กรอบโต้ตอบ Define Name จะปรากฏ

4. ที่ช่อง Name เปลี่ยนชื่อให้เป็น Beeper ที่กรอบมาโคร คลิ๊กเม้าส์ที่หน้าข้อความ Command จากนั้นกำหนด Shortcut Key เป็น b ดังแสดงในรูปที่ 13

5. คลิ๊ก OK



รูปที่ 13 การตั้งชื่อให้มาโคร Beeper

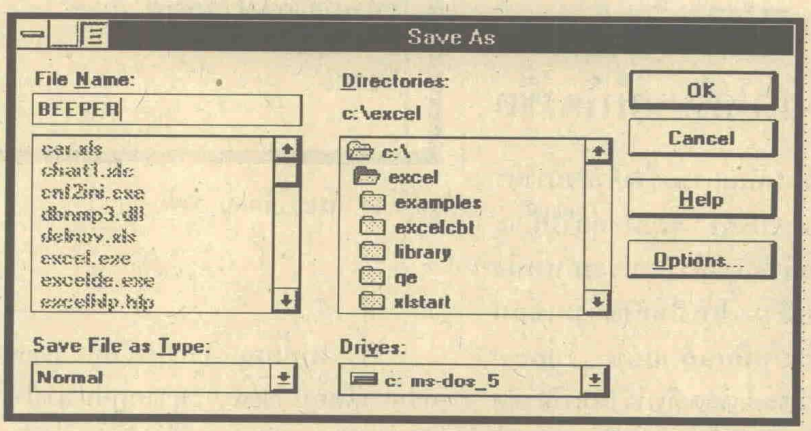
ทดสอบรันมาโคร Beeper

1. ไปที่เวอร์กชีต
2. กด Ctrl + b

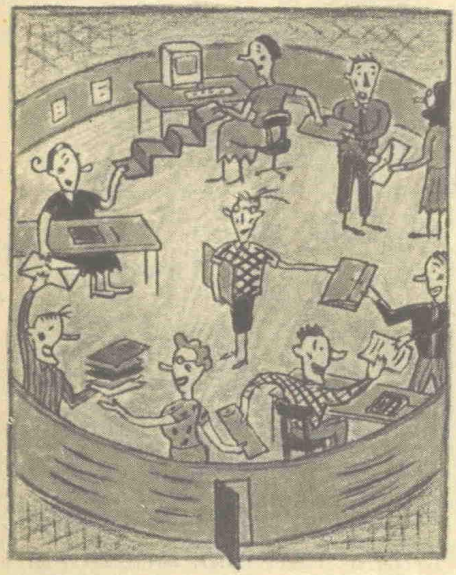
ท่านจะได้ยินเสียง Beep สั้น ๆ

การเซฟมาโคร

เราจะเซฟมาโคร Beeper ซึ่งขณะนี้ควรจะอยู่ในมาโครชีตชื่อ Macro1 ถ้าไม่ใช่ไม่ต้องตกใจ เพราะว่าเราจะเซฟลงไฟล์ชื่อ BEEPER ขั้นตอนมีดังนี้



รูปที่ 14 ที่กรอบ File Name พิมพ์ BEEPER



1. ไปที่มาโครชีตที่เก็บมาโคร Beeper (ของผู้เขียนคือ MACRO1.XLM)
2. ไปที่เมนู File เลือก Save As...
3. ที่กรอบ File Name พิมพ์ BEEPER ดังรูปที่ 14
4. คลิ๊ก OK

เนื้อหาตอนต่อไป

- แนวทางในการเขียนมาโครให้มีประสิทธิภาพ
- การจัดฟอร์แมตให้กับมาโคร
- หมวดคำสั่งพื้นฐาน
- การประยุกต์ใช้คำสั่งเงื่อนไข
- การประยุกต์ใช้ฟังก์ชันที่จำเป็นในการเขียนมาโคร